

Miguel Isaac Pérez Peña

Masivian Árbol binario

Instrucciones:

Prerrequisitos:

Obligatorio: IISExpress, .net core 2.1, visual studio 2017+

Opcional: Postman, WebSurge.

Instalación

La solución puede ser encontrada en la siguiente URL:

<https://github.com/MiguelIPerez/BinaryTree>

O puede ser clonada usando el siguiente Link:

<https://github.com/MiguelIPerez/BinaryTree.git>

Una vez descargada la solución, abrir el archivo: 'BinaryTree.sln'.

Nota: Todo el código esta en ingles dado que he trabajado mucho tiempo con gente de otros países y estoy muy acostumbrado a escribirlo así y ahora me cuesta escribirlo en español.

Una vez abierto se encontraran 3 proyectos:

- BinaryTreeAPI – Donde reside el API Rest.
- BinaryTreeDomain – Donde reside la lógica del proyecto.
- BinaryTreeTests – Donde Residen las pruebas unitarias.

Ejecución:

Pruebas Unitarias:

Una vez abierto la solución dentro de visual studio. Ir a View -> Solution Explorer:

Debería aparecer algo como esto:

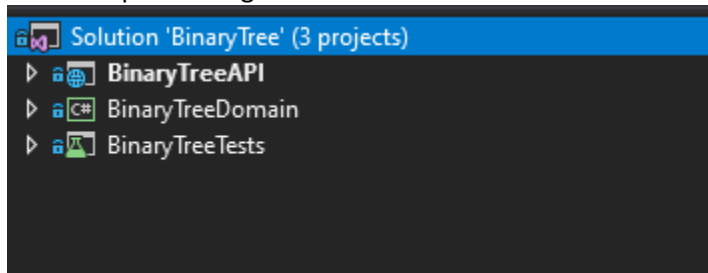
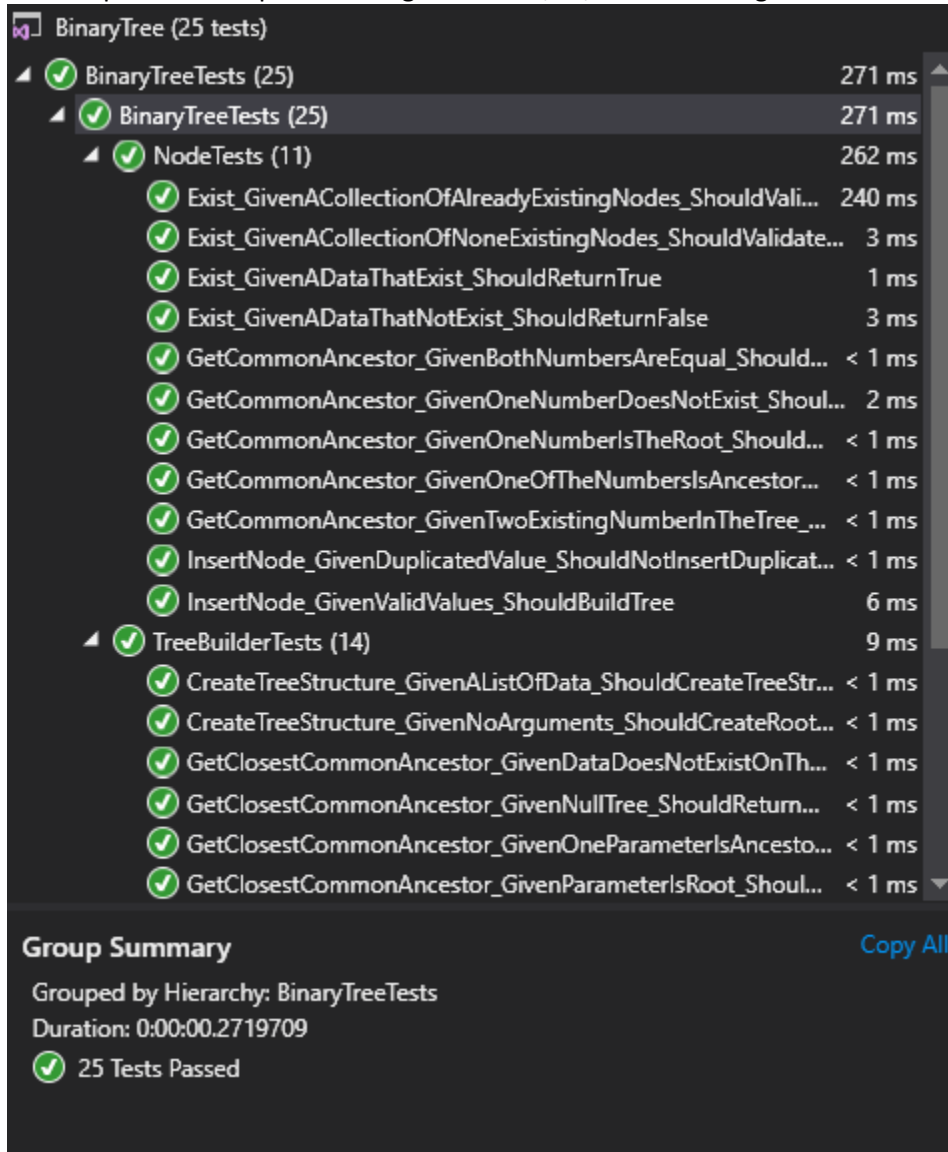


Figure 1 vista global de la solución

Luego seleccionar el nodo raíz llamado "Solution 'BinaryTree (3 projects)'" click derecho -> build.

Una vez la compilación o el "build" haya terminado, desplegar el nodo llamado 'BinaryTreeTests' donde se encontrara los test unitarios relacionados a los objetos del dominio que tienen la lógica del árbol binario.

Para ejecutarlos simplemente vaya a la barra de herramientas y seleccione "Test" -> "Windows" -> "Test Explorer" o simplemente digite "Ctrl +E, T", deberá ver algo como esto:



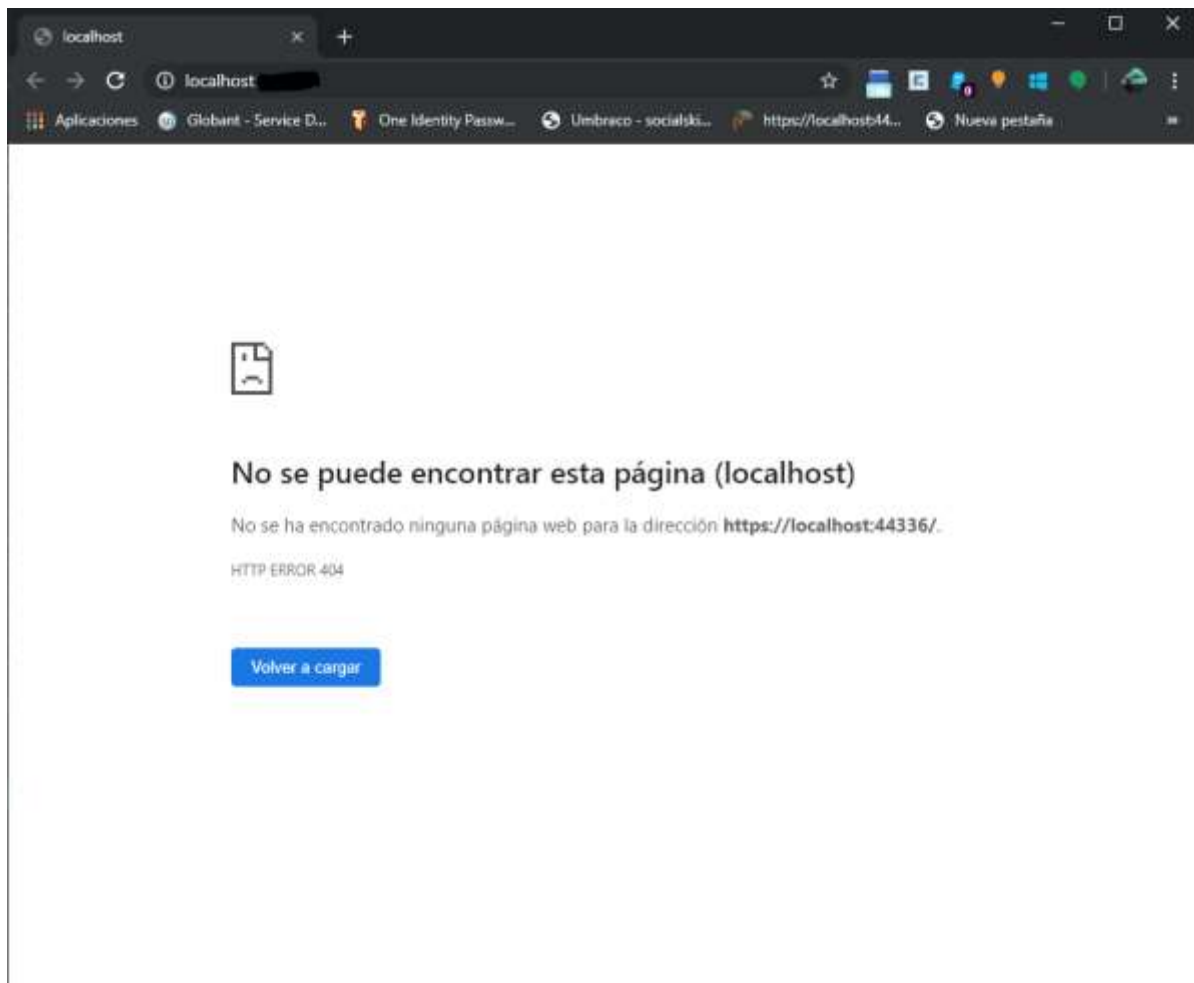
Parece sobre el nodo resaltado en la imagen, "Click derecho" -> "Run Selected Test".

Todos los test unitarios deberían ser ejecutados y exitosos.

API Rest.

Vaya a la barra de herramientas seleccione "Debug" -> "Start debugging" para depurar el código fuente (menor desempeño) o "Debug" -> "Start without debugging" ejecutara la solución por fuera del ambiente de visual studio (mucho mejor desempeño pero no podrá depurar el código fuente).

Una vez ejecutado el paso anterior deberá aparecer una ventana de navegador de internet así:



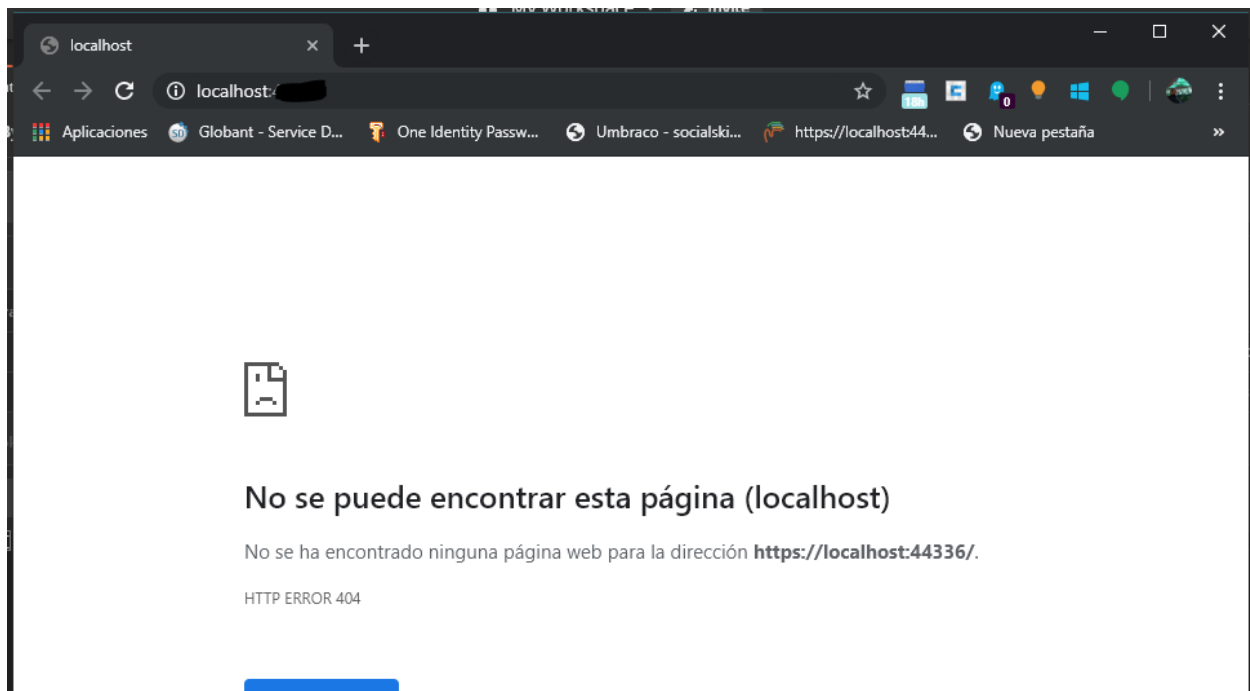
No es un error, es simplemente que el navegador no encuentra una página que mostrar, pero el API Rest está activo y funcionando.

Probar el API Rest:

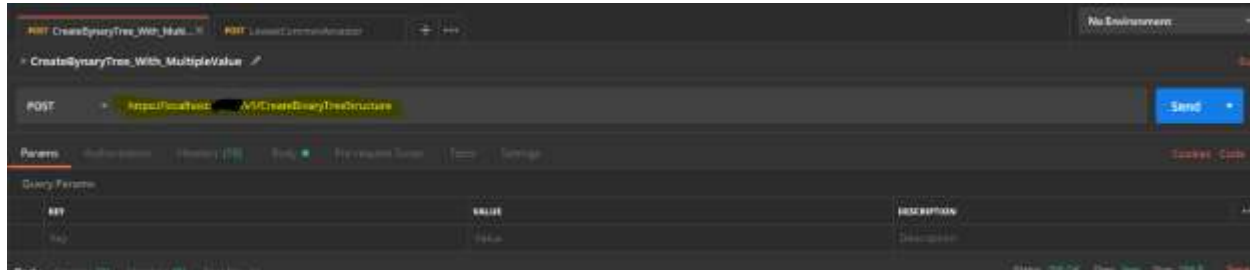
Si tiene Postman o puede tener acceso a él, he agregado a la carpeta de esta solución un archivo llamado "BinaryTree.postman_collection" que tiene ejemplos claros de cómo llamar a los dos Endpoints de dicha solución.

Simplemente dentro de Postman importe el archivo mencionado y tendrá dos ejemplos claros de como funciona la solución.

Deberá cambiar el puerto de acuerdo a la información que le muestre el navegador de la siguiente manera:



Copie la URL que ve en el navegador incluyendo los números que están después así <https://localhost:xxxxx> y luego reemplace en Postman en la barra de direcciones solo la parte que dice así: <https://localhost:xxxxx>, el resto de la URL deberá continuar como esta:



Este archivo ya contiene los headers y el body necesario para hacer los request de manera satisfactoria.

Nota: Para el endpoint : “CrearArbolBinario” estuve confundido si tenía que retornar un árbol con valores estáticos o darle la oportunidad al cliente del servicio que mandara información para la creación de un árbol personalizado, así que opté por dejar que la creación fuese dinámica, y así cliente en el body del request solo tendría que mandar un json con una propiedad llamada DataList que es un arreglo de enteros y con eso sería suficiente para crear un árbol dado que el primer valor del arreglo será considerado el Nodo Raíz el resto se puede distribuir como nodos descendentes de acuerdo a la lógica implementada.

Probar API con carga de trabajo.

Si tiene WebSurge o tiene la opción de descargarlo, también cree un archivo llamado "BinaryLoadTest.websurge" el cual podrá ser usado dentro de la aplicación para crear multiples llamados en un determinado lapso de tiempo

Dicho archivo ya viene configurado para lanzar aproximadamente 10000 peticiones en un minuto (porque es lo máximo que puede aguantar el pc en el que estoy trabajando), pero con pequeñas configuraciones podría lanzar más.

Espero sea de su agrado y podamos llegar a conversar

MP